

Article

Tracking of a Fixed-Shape Moving Object Based on the Gradient Descent Method

Haris Masood ¹, Amad Zafar ², Muhammad Umair Ali ³, Tehseen Hussain ¹, Muhammad Attique Khan ⁴, Usman Tariq ⁵ and Robertas Damaševičius ^{6,*}

- ¹ Electrical Engineering Department, Wah Engineering College, University of Wah, Wah Cantt 47040, Pakistan; haris.masood@wecuw.edu.pk (H.M.); tehseen.hussain@wecuw.edu.pk (T.H.)
- ² Department of Electrical Engineering, The Ibadat International University, Islamabad 54590, Pakistan; ammad_zaffar@yahoo.com
- ³ Department of Unmanned Vehicle Engineering, Sejong University, Seoul 05006, Korea; umair@sejong.ac.kr
- ⁴ Department of Computer Science, HITEC University Taxila, Taxila 47040, Pakistan; attique.khan@hitecuni.edu.pk
- ⁵ College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; u.tariq@psau.edu.sa
- ⁶ Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania
- * Correspondence: robertas.damasevicius@vdu.lt

Abstract: Tracking moving objects is one of the most promising yet the most challenging research areas pertaining to computer vision, pattern recognition and image processing. The challenges associated with object tracking range from problems pertaining to camera axis orientations to object occlusion. In addition, variations in remote scene environments add to the difficulties related to object tracking. All the mentioned challenges and problems pertaining to object tracking make the procedure computationally complex and time-consuming. In this paper, a stochastic gradient-based optimization technique has been used in conjunction with particle filters for object tracking. First, the object that needs to be tracked is detected using the Maximum Average Correlation Height (MACH) filter. The object of interest is detected based on the presence of a correlation peak and average similarity measure. The results of object detection are fed to the tracking routine. The gradient descent technique is employed for object tracking and is used to optimize the particle filters. The gradient descent technique allows particles to converge quickly, allowing less time for the object to be tracked. The results of the proposed algorithm are compared with similar state-of-the-art tracking algorithms on five datasets that include both artificial moving objects and humans to show that the gradient-based tracking algorithm provides better results, both in terms of accuracy and speed.

Keywords: object recognition; object tracking; gradient descent; particle filters

Citation: Masood, H.; Zafar, A.; Ali, M.U.; Hussain, T.; Khan, M.A.; Tariq, U.; Damaševičius, R. Tracking of a Fixed-Shape Moving Object Based on the Gradient Descent Method. *Sensors* **2022**, *22*, 1098. <https://doi.org/10.3390/s22031098>

Academic Editors: Brij B Gupta and Alicia García-Holgado

Received: 30 December 2021

Accepted: 28 January 2022

Published: 31 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Object recognition and tracking is still a major area of interest when it comes to digital image processing, pattern recognition, convolution neural networks and artificial intelligence [1]. The applications associated with object recognition range from surveillance [2], optical character recognition [3], human behavior detection [4], remote sensing [5], video activity localization [6], night-time vision [7] and biomedical image acquisition applications to deep learning techniques [8]. Although many applications have been developed thus far, the need for the optimization of the algorithms in terms of convergence and time minimization still persists. Object tracking particularly has many unique challenges associated with it, such as dealing with variations in scaling [9], occlusion [10], shift [11], camera axis orientations [12], etc.

Training tracking algorithms, such as approximate proximal gradient methods [13] and

rapid gradient descent [14], is, in general, a very complex optimization issue because it involves a large number of secondary variables. Depending on the datasets and the problem at hand, the goal is to implement a tracking routine that provides faster results compared to its predecessors [15]. Besides the faster results, the algorithm should be accurate enough to concentrate on the object of interest only by minimizing the average tracking error. During the tracking routine, the tuning and assignment of weights are of utmost importance since they are the ones that mostly result in accurate prediction and estimation processes. For this purpose, a deep neural network technique known as gradient descent has been employed, which focuses on setting the weight of the parameters based on the lowest loss function.

Typically, object tracking is associated with several state-of-the-art techniques that are based on deep neural networks, artificial neural networks and convolution neural networks (CNNs) [16,17]. All of the aforementioned techniques have limitations ranging from poor interpretation and recognition of the object of interest to structural design issues. To solve these issues and to enhance the convergence of the algorithms, gradient descent training algorithms were proposed [18,19]. The gradient descent algorithms have the tendency to overcome most of the shortcomings of their predecessors by quickly converging into local minima but in an efficient manner [20].

One of the most famous types of gradient descent techniques is known as stochastic gradient descent (SGD). SGD is known to combine the benefits of basic gradient descent algorithms, i.e., the stochastic strategy and the backpropagation [21]. SGD can be used for image processing applications as it uses the backpropagation to converge quickly using local minima. This enables tracking algorithms such as particle filters to track the object in an efficient manner by quickly converging towards the object of interest [22–25].

In the previous decade, lots of visual tracking methods have been implemented, each having its share of pros and cons. The visual tracking methods can be categorized into two main classes, i.e., discriminative algorithms and generative algorithms. The generative methods are known to classify the object of interest by convolving it with a kernel first, followed by the tracking process that selects the most suitable candidate with an appearance model most suited to the chosen template. The most popular generative methods are particle-Kalman filters [26], Kalman filters [27], kernel-based object tracking [28], etc. On the contrary, the discriminative methods use binary classifiers in order to discriminate the object of interest from the background. The most popular discriminative methods employed thus far are ensemble tracking methods [29] and LDA and Bayes inference methods [30].

In [31], first, the dynamic behavior of the tracking model was assumed to be linear, which was used to model the motion of the objects using the parametric single acceleration method. The two sub-model states are estimated using an H filter. The estimates then act as input to the particle filters, resulting in the optimized state. The local estimates are mixed with the proposed interactive model for the calculation of the posterior location of the object of interest. Shi et al. [32] employed sparse representation for modeling of the target object. The target localization problem was assumed to be an L1 norm-related minimization problem and was resolved using convex optimization. The method was further improved [33] by proposing an lp regularization model. The lp regularization model was minimized using the accelerated proximal gradient approach, which ensured rapid convergence and less average tracking errors as compared to its predecessors [34]. The conditional random field (CRF) model has been used to combine multiple image texture, shape, context and location features for multiclass object recognition [35].

An eigenspace model for object tracking employs feature vectors linked with pixels in the target template, which are regarded as discrete observations of the target object [36]. To arrive at an eigenspace representation, the collection of observations is trained via non-linear subspace projection. A similarity function in the eigenspace representation is used to perform localization and segmentation. To optimize the similarity function regarding the transformation parameters, gradient descent and mean-shift approaches are used.

In this paper, an optimized algorithm for object recognition and tracking is proposed using SGD. The aim is also to develop a tracking routine that is able to track the

object of interest under different scenarios. The proposed tracker will be tested on the target objects which undergo changes in appearance, scale and camera axis orientations. First, the object of interest is detected using the maximum average correlation height (MACH) filter. The object is detected using different parameters of the MACH filters. The detected object is then tracked in successive video frames using the SGD-based particle filters, which are enhanced forms of the conventional particle filter, providing better convergence than their predecessors. A comparison with similar state-of-the-art algorithms is performed to prove the effectiveness of the algorithm.

2. Proposed Methodology

The proposed methodology followed for the implementation of the algorithm is shown in Figure 1. The main algorithm can be split up into three parts, i.e., preprocessing, object recognition and object tracking. The trained images are kept in the library for object recognition purposes. Once the testing images are obtained, they are first fed to the preprocessing block to cater for any noise and/or any smoothing and sharpening abnormalities.

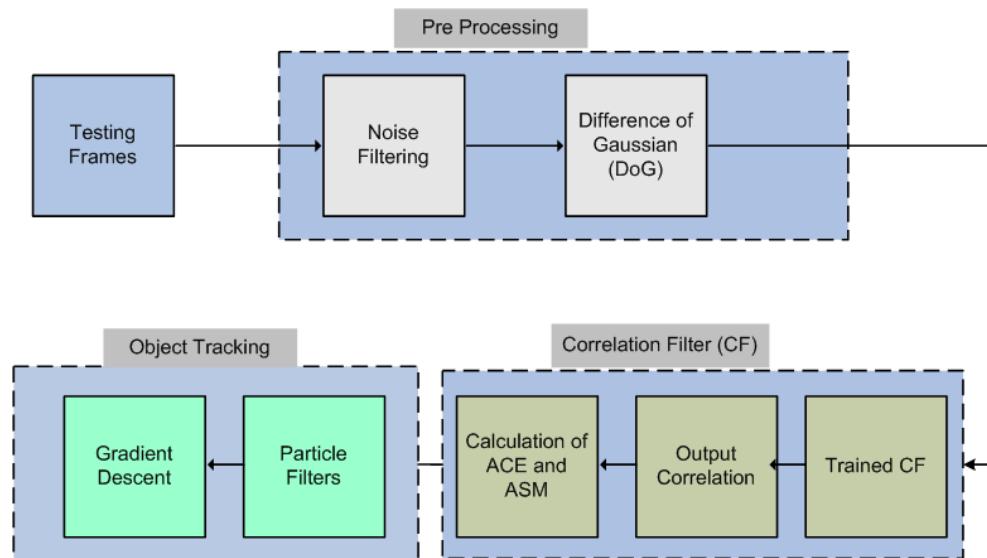


Figure 1. Proposed model of a system.

2.1. Preprocessing

Preprocessing is considered one of the most fundamental and very important steps in any image processing application. The step ensures that all the images possess similar dimensions and properties before the actual algorithm is applied to them. For preprocessing of the images, the difference of Gaussian (DoG) has been applied before the actual preprocessing of the images. The DoG filter not only reduces the noise by applying the Gaussian motion blur but also enhances the edges, which can be considered a major advantage when it comes to an image processing application [37]. The DoG simply is a two-step process. First, the DoG performs edge detection by applying the Gaussian motion blur. The motion blurring allows the removal of any unwanted noisy components by applying the smoothing process. The algorithm then applies another motion blur using a sharper theta. As the name implies, the final image is made by replacing each image pixel with the difference of blurred images as shown in Figure 2.

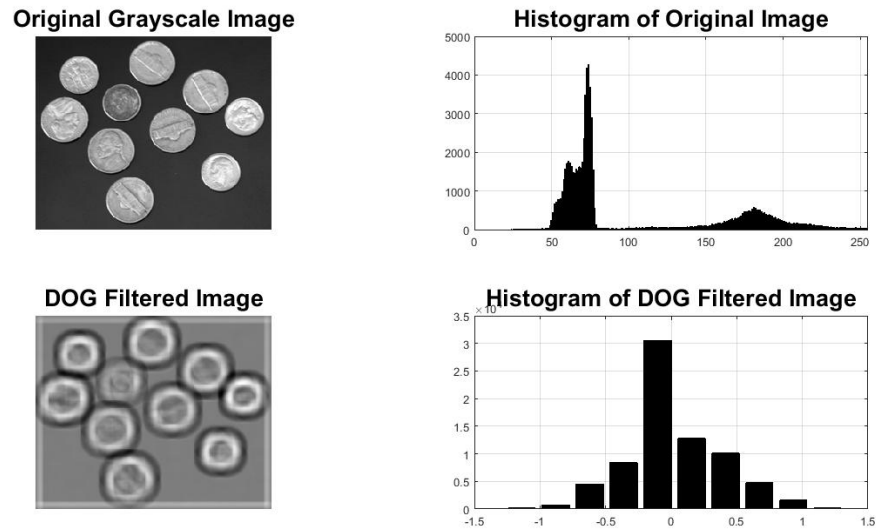


Figure 2. Visual explanation of the difference of Gaussian (DoG) method.

The DoG is a bandpass filter that is used to compute wavelets that are symmetric in nature. By changing the standard deviation in the equation of DoG, the bandpass frequency can be altered. The value of the bandpass frequency must be selected so that it can provide the best tradeoff between intra-class distortion and inter-class discrimination. Combining DoG with the MACH filter results in much sharper correlation peaks because of the built-in tendency of DoG to detect edges. The DoG filter enhances the edges by approximating the Mexican hat wavelet. The DoG is actually a variance between two scaled Gaussian functions $g_i(x, y)$, where $i = 1, 2, \dots$ [38]:

$$g_1(x, y) = \frac{1}{2\pi\delta_1^2} \cdot \exp\left(-\frac{x^2 + y^2}{2 \cdot \pi \cdot \delta_1^2}\right) \quad (1)$$

Similarly, for $i = 2$,

$$g_2(x, y) = \frac{1}{2\pi\delta_2^2} \cdot \exp\left(-\frac{x^2 + y^2}{2 \cdot \pi \cdot \delta_2^2}\right) \quad (2)$$

The DoG filter is applied by taking the difference between Equations (1) and (2).

$$g(x, y) = g_1(x, y) - g_2(x, y) \quad (3)$$

By combining Equations (1)–(3), we obtain

$$g(x, y) = \frac{1}{2\pi\delta_1^2} \cdot \exp\left(-\frac{x^2 + y^2}{2 \cdot \pi \cdot \delta_1^2}\right) - \frac{1}{2\pi\delta_2^2} \cdot \exp\left(-\frac{x^2 + y^2}{2 \cdot \pi \cdot \delta_2^2}\right) \quad (4)$$

The rule of thumb for a successful DoG application is to choose the bandpass of the Mexican wavelet as the ratio of δ_1 and δ_2 . Experimental results have shown that DoG yields closest approximation if the ratio of δ_1 and δ_2 is 1.6.

2.2. Object Recognition

Object recognition has been performed using the MACH filter, which is a class of correlation filters. The identification of an object is a straightforward approach, but identifying an object in the case of change in camera axis orientations or in the presence of noise is a challenging task. The correlation filters are chosen because of their ability to detect and identify the object of interest under challenging circumstances. Another advantage of correlation filters is they are computationally less expensive as compared to their counterparts. The MACH filter uses a set of training images for the computation of

correlation peaks. The MACH distinguishes the training images into true and false classes, where the true class of training images depicts the set of images that the filter retains, while the false class represents the discarded set of images.

The MACH filters are known for their ability to suppress noise as well. The noise suppression ability comes with their ability to maximize average similarity measure (ASM) and reduce distortion. The MACH filter is based on enhancing the four main parameters, i.e., average correlation energy (ACE), average similarity measure (ASM), output noise variance (ONV) and average correlation energy (ACE). However, the most important aspect is the ASM, which is directly associated with the correlation peak. As mentioned above, the energy equation associated with MACH is based on four parameters.

$$E(f) = \alpha(ONV) + \beta(ACE) + \gamma(ASM) - \delta(ACH) \quad (5)$$

For the MACH implementation, Equation (5) needs to be minimized. The “T” sign in superscript represents the transpose of the function

$$\alpha f^T C f + \beta f^T D_x f + \gamma f^T S_x f - \delta f^T m_x \quad (6)$$

As stated earlier, MACH is known to recognize the object of interest by minimizing the ACE and ASM while maximizing the average correlation height. The minimization of ASM is achieved using Equation (7). “*” represents the complex conjugate of the function.

$$ASM = h^+ \left[\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^* (X_i - \bar{X}) \right] h = h^+ S_x h \quad (7)$$

where the similarity of the training images is represented by the matrix S. Similarly, the minimization of ACE is achieved using Equation (8)

$$ACE = h^+ \left(\frac{1}{N} \sum_{i=1}^N X_i X_i^* \right) h = h^+ D_x h \quad (8)$$

$$|\bar{u}|^2 = |h^+ \bar{x}|^2 = h^+ \bar{x} \bar{x}^+ h \quad (9)$$

The average correlation intensity, which is depicted by $|\bar{u}|^2$, is maximized by the filter h . The same h is also used for the minimization of ASM and ACE.

While minimizing Equation (5), Equation (10) can easily be extracted after calculating the ASM and ACE.

$$f = \frac{m_x^*}{\alpha C + \beta D_x + \gamma S_x} \quad (10)$$

The three parameters mentioned in the denominator of Equation (10) are considered crucial in the performance of MACH. The optimal value selection of these parameters is called optimal tradeoff (OT), which was initially proposed by Bone et al. [39]. The OT values of the parameters are based on two parameters, i.e., peak-to-correlation energy (PCE) and correlation output peak intensity (COPI) [40,41]. The PCE and COPI are calculated using Equations (11) and (12).

$$COPI = \max\{|C(x, y)|^2\} \quad (11)$$

$$PCE = \frac{COPI - \overline{|C(x, y)|^2}}{\left\{ \sum \frac{[|C(x, y)|^2 - \overline{|C(x, y)|^2}]^2}{N_x N_y - 1} \right\}^{1/2}} \quad (12)$$

where $\overline{|C(x, y)|^2} = \sum |C(x, y)|^2 / N_x N_y$ is the average correlation output intensity.

Bone et al. [39] suggested that the values of 0.01, 0.1 and 0.3, respectively, for α , β and γ for the COPI cost function may be considered optimal.

Figure 3 shows a Blurred Vehicle along with the MACH filter results. It is pertinent to note here that even though the vehicle is undergoing severe blurriness, MACH is still able to detect the object of interest using the PCE and COPI indexes. The PCE and COPI values achieved for the object mentioned in Figure 3 are 2.3047×10^{-5} and 32.1376.

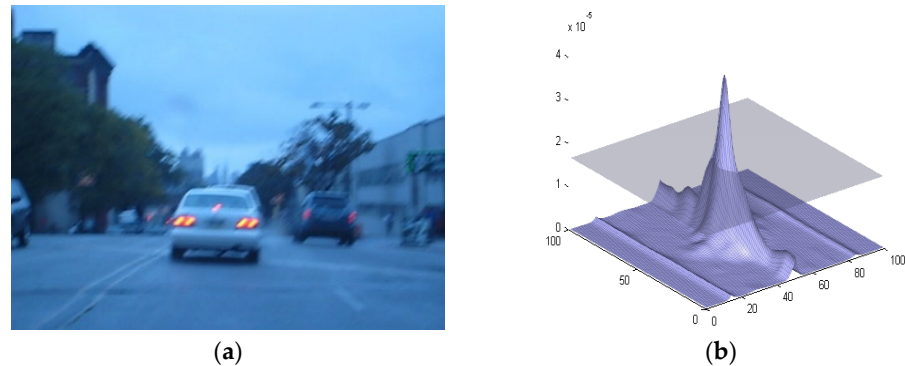


Figure 3. (a) A sample image of Blur Car (Data Set-1) and (b) MACH results.

As mentioned earlier, MACH has the ability to detect the object of interest even if it undergoes changes in scaling, shifting or camera axis orientations. Figure 4 shows a vehicle traveling at night, and it has been scaled out by a factor of 3. MACH can still extract the correlation peak with a PCE value of 29.21.

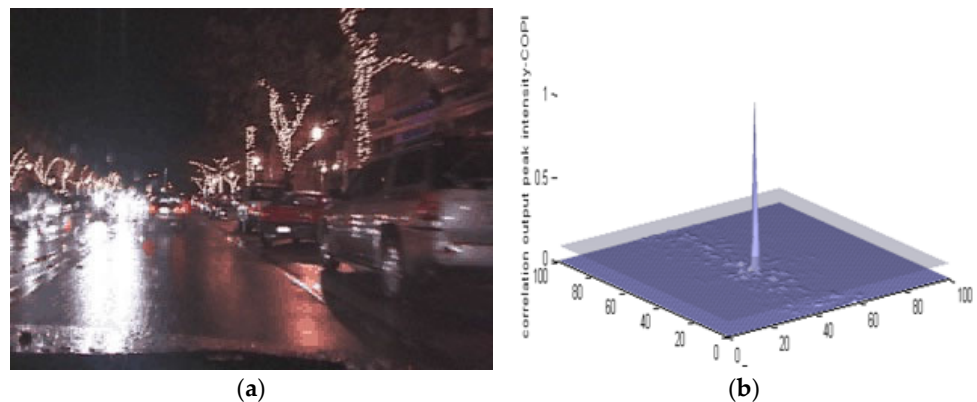


Figure 4. (a) Vehicle traveling at night (Data Set-3) (b) MACH results.

Figure 5 shows a dog in a running mode along with its MACH results. The dog is the object of interest, which needs to be detected using MACH. The dog is currently encountering in-plane rotation and is not showing its original physical attributes. Even in such a tricky case like this, the MACH can detect the object of interest using a PCE value of 66.01.

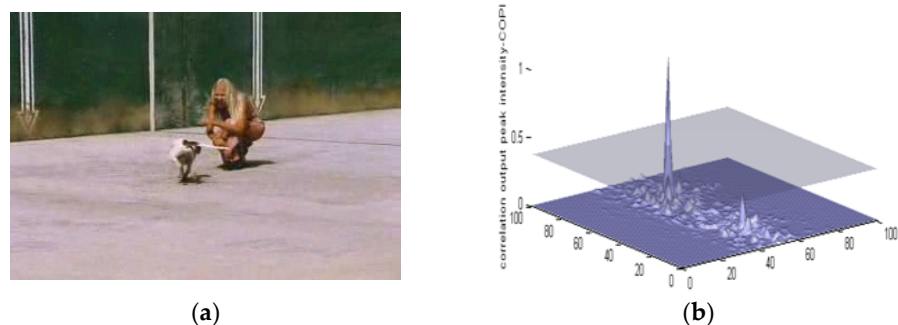


Figure 5. (a) A sample image of a Running Dog (Data Set-2) and (b) MACH results.

Figure 6 shows a partially occluded case of a vehicle traveling on a road along with its MACH results. Besides the natural attributes of MACH providing invariance against shift, scales and camera axis orientations, it also possesses the ability to detect an object even it is partially occluded. Figure 6 shows that even the vehicle is partially occluded, MACH is still able to detect the object using PCE and COPI values of 68.5701 and 0.0014, respectively. The object of interest in this case was 30% occluded, and MACH still detects it successfully.

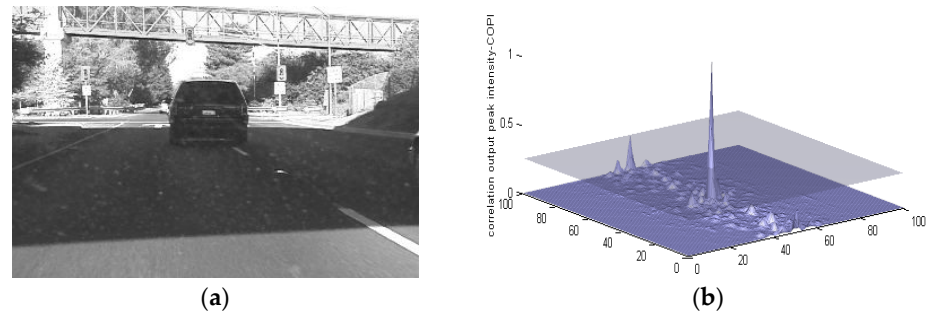


Figure 6. (a) An occluded grayscale vehicle (Data Set-4) and (b) MACH results.

Once the object of interest is detected using MACH, a bounding box is used to encapsulate the object of interest, as mentioned in Figure 7. The coordinates of the bounding box are then fed to the gradient descent-based object tracking routine, which will update the bounding box with respect to the apparent motion of the object.

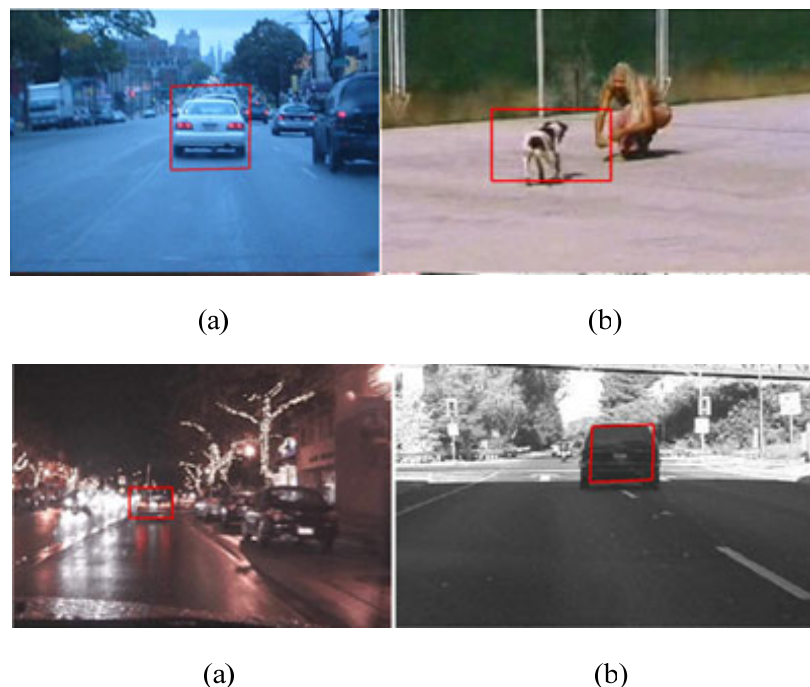


Figure 7. Detected objects for sample images: (a) Blurred Vehicle; (b) Running Dog; (c) Vehicle at Night; (d) Car moving in a lane.

2.3. Object Tracking

The goal of object tracking is to track the recognized object of interest in successive frames. The main object tracking algorithm employs the gradient descent-based particle filters. The gradient descent technique allows the particle filters to track the object of interest in less time as compared to the conventional particle filters.

Gradient Descent-Based Particle Filtering

Gradient descent technique is one of the most famous methods used for optimizing algorithms. They have the natural tendency to converge when used together with other deep learning-based algorithms. In this paper, the gradient descent technique is used in conjunction with particle filters in order to improve the efficiency of the particle filters. The gradient descent technique optimizes the algorithms by working on their loss functions. The loss function can be described as the apparent difference between the function output and the samples. For this purpose, a hypothesis function can be assigned to a linear regression mathematical model, such as $h_{\theta}(x) = \theta_0 + \theta_1 x$, where θ_0 and θ_1 defines the equation factors. The samples can be defined as vector $(x_i, y_i) = (1, 2, \dots, n)$, such that every input x_i corresponds to an output y_i . The loss function generally can be defined using Equation (13).

$$J(\theta_0, \theta_1) = \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2 \quad (13)$$

The optimization of the loss function and the mathematical model is the main goal of the gradient descent technique. This means that the gradient descent technique can be used to amend or modify the mathematical model of a particle filter by reducing the eventual loss function.

In the case of particle filters, the hypothesis function can be very complex. Therefore, there is a need to define a more complicated hypothesis function. Adding more factors to the hypothesis function makes it more complex such that $h_{\theta}(x_1, x_2, \dots, x_n) = \theta_1 + \theta_1 x_1 + \dots + \theta_n x_n$. The loss function defined in Equation (13) can be made more complex using Equation (14).

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=0}^m (h_{\theta}(x_0, \dots, x_n) - y_i)^2 \quad (14)$$

To minimize the cumulative loss function based on the ensemble of particle filters and the gradient descent technique, partial derivatives can be applied. Gradients are generically applied for measuring the trend of the aim function. Therefore, it is possible to associate the θ_i result with the loss function. Equation (15) shows how a loss function can be represented using a gradient.

$$\frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1, \dots, \theta_n) \quad (15)$$

The precision of the function, i.e., the difference between the samples and the mathematical model, is represented by ϵ , which is known as the terminal function. The estimation process of particle filters will stop when the estimated difference becomes less or equal to ϵ . The particle filters are based on the prediction, estimation and upgrade regimes. The percentage of gradient used to upgrade the state of particle filters is determined by the controlling parameter α , which is the step size. The amended update expression for particle filters becomes:

$$\theta_{iold} - \alpha \frac{\partial}{\partial \theta_i} J(\theta_1, \dots, \theta_n) \quad (16)$$

Since the function pertaining to the particle filters is convex in nature, it can be optimized using gradient descent. Another important aspect is the selection of the step size. Selecting a smaller step size may result in identifying the most optimal solution, but the downside is a very slow convergence speed. On the contrary, a larger step size may result in an elevated speed but does not ensure an optimal solution.

Considering the pros and cons of both approaches, in this paper, the gradient descent technique is used for updating the probabilities and assigned weights to the particles using the steps mentioned in Algorithm 1. The conventional particle filtering process

involves the states $p(X_n|Y_{1:n})$. We employ particle filters in combination with the gradient descent technique because the particle filters can work on both the linear and nonlinear systems, while gradient descent allows the particles of the particle filters to converge quickly. The quick convergence allows a much quicker tracking process as compared to the conventional particle filtering routines. Gradient descents' ensemble with the particle filters starts by assigning a weight to the samples.

Algorithm 1: Gradient Descent Algorithm

1. Initializing the aim function parameters: $\theta_0, \theta_1, \dots, \theta_n$;
 2. Initialize step size α and terminal distance of recursion parameter ϵ ;
 3. Gradient calculation of the function using the loss functions' partial derivative using Equation (15);
 4. Apply the gradient descent algorithm on particle filters;
 5. If the gradient descents, i.e., $J(\theta_0, \theta_1, \dots, \theta_n) \leq \epsilon$, stop the converging process;
 6. Otherwise, continue the process;
 7. Employ the gradient descent process by multiplying the step size α with the gradient;
 8. Renew all the values of θ using Equation (17).
-

Using the steps (5) and (6) of the gradient descent algorithm, new weights are assigned to the particles using Equation (17).

$$W_n^i = W_{n-1}^i \frac{p(Y_n|X_n^i)p(X_n^i|X_{n-1}^i)}{\Pi(X_n|X_{n-1}^i, Y_n)}, \sum_{i=1}^N W_n^i = 1 \quad (17)$$

Equation (17) shows the working particle filters after the implementation of gradient descent. The particle filters are based on two steps, i.e., prediction and then update. Equation (17) basically merges the two steps. The part of the equation involving the $p(Y_n|X_n^i)$ is used for the implementation of the probabilistic model. Once the probabilities are calculated, the weights of particle filters are set according to the expression $\{X_{n-1}^i, W_{n-1}^i\}$. However, in contrast to the conventional particle filters concept, in this paper, the gradient descent technique is used to update the weights according to Algorithm 1.

3. Results and Discussion

Five different data sets have been used for testing the results of the gradient-based tracking routine. Since the gradient descent technique is known as the optimization algorithm, the results of the proposed algorithm will be compared with other state-of-the-art algorithms.

3.1. Data Sets

The first employed data set is the Blur Car, which shows a white vehicle moving on a road [42]. The data set is challenging in the sense that it possess challenges related to camera axis orientations as well as blurriness of the object of interest. Figure 8 shows the results of applying the gradient descent-based tracking routine. The tracker successfully tracks the vehicle in successive frames.

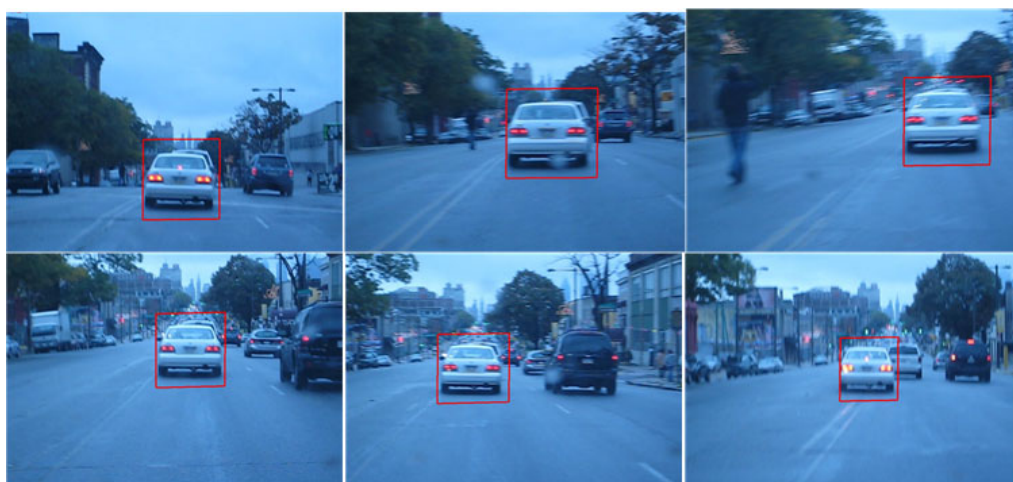


Figure 8. Tracking of Vehicle (Data Set: Blur Car).

The second data set is Running Dog, which shows a dog running on a floor [43]. The data set is challenging in the sense that the object is changing shape. In addition, the data set also shows blurriness of the object of interest. Figure 9 shows the results of applying the gradient descent-based tracking routine. The tracker successfully tracks the vehicle in successive frames.

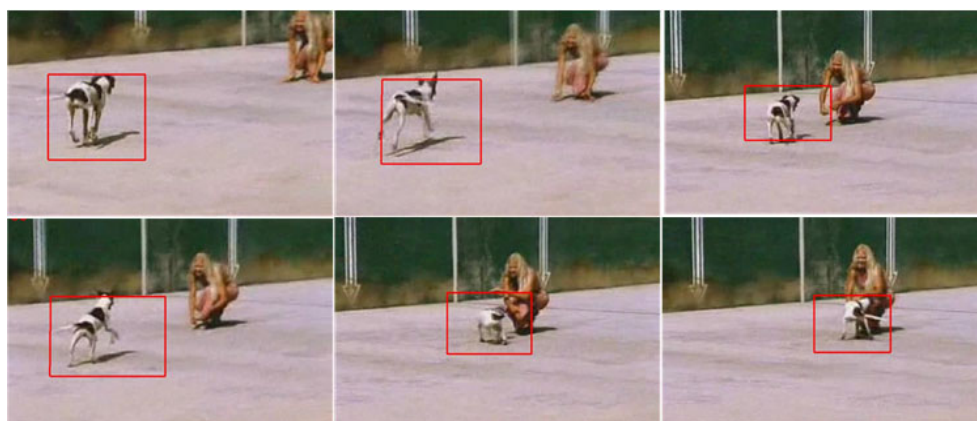


Figure 9. Tracking of a running dog (Data Set: Running Dog).

The third data set is Vehicle at Night, which shows a vehicle travelling at night [44]. The data set is challenging in the sense that the object is traveling at night under poor illumination conditions. In addition, the data set also shows blurriness of the object of interest. Figure 10 shows the results of applying the gradient descent-based tracking routine. The tracker successfully tracks the vehicle in successive frames.



Figure 10. Tracking of Vehicle (Data Set: Vehicle at Night).

The fourth data set is a grayscale vehicle traveling on the road [43]. The data set is challenging in the sense that the object becomes occluded under a bridge. In addition, the data set also shows blurriness of the object of interest. Figure 11 shows the results of applying the gradient descent-based tracking routine. The tracker successfully tracks the vehicle in successive frames.

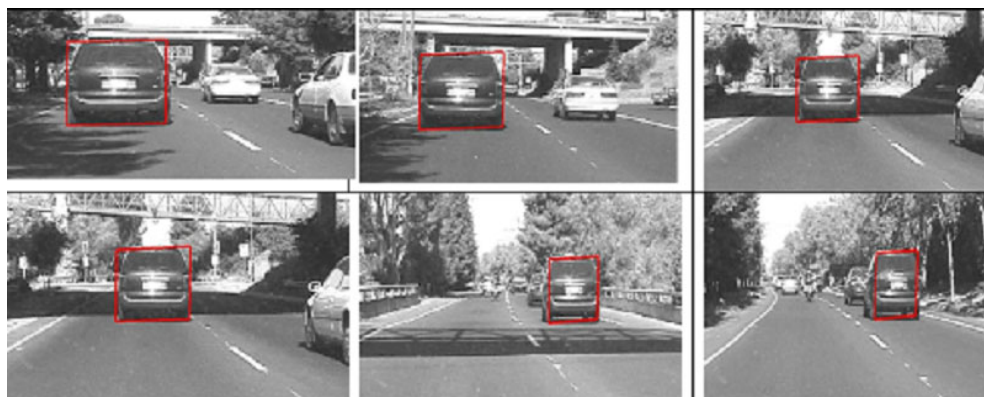


Figure 11. Tracking of a grayscale occluded vehicle (Data Set: Grayscale Vehicle).

The fifth data set utilized is called “Singer” [13], and it contains aphotographs of a singer performing in a concert. The dataset is deemed significant since the pictures are continually zoomed in and out, posing a challenge to the object tracking system, as illustrated in Figure 12.

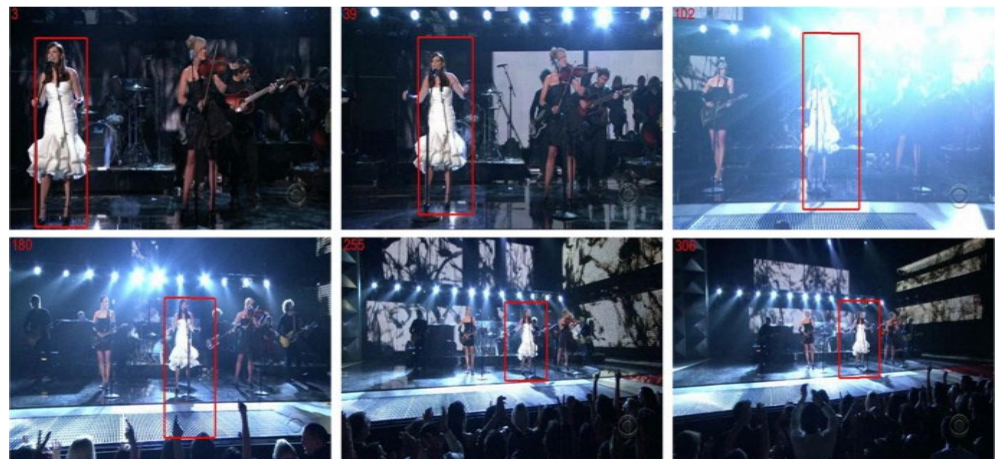


Figure 12. Tracking of a human person in a complex environment (Data Set: Singer).

3.2. Discussion and Comparison

To prove the efficiency of the algorithm with other state-of-the-art similar algorithms, the gradient-based tracking technique has been compared with four other recently proposed techniques. The algorithm has been compared with a target tracking algorithm based on Convolution Neural Network (TTACNN) [44], ADT: object tracking algorithm. Based on adaptive detection [45], vehicle tracking algorithm combining detector and tracker (VTACDT) [46], multi-object tracking for urban and multilane traffic (MTUMT) [47], adaptive weighted strategy and occlusion detection mechanism (AWSODM) [48] and approximate proximal gradient-based correlation filter (APGCF) [13]. Table 1 shows the execution times of these algorithms, while Table 2 represents the average tracking errors.

Table 1. Comparison of state-of-the-art algorithms in terms of execution time (sec.).

Comparison of Execution Time (in Seconds) of Algorithms (Min. 300 Frames)							
Data Set	TTACNN	ADT	VTACDT	APGCF	AWSODM	MTUMT	Proposed Algorithm
Blur Car	2.14	2.51	2.10	2.44	2.91	2.29	2.01
Running Dog	2.92	4.12	2.77	4.11	2.99	2.84	2.89
Vehicle at Night	3.04	3.09	2.91	3.19	2.71	2.69	2.72
Grayscale vehicle	2.46	3.01	2.62	2.90	2.19	2.90	2.21
Singer	2.99	3.71	2.81	2.81	2.89	3.11	2.85

Table 2. Comparison of different techniques based on Average Tracking Errors.

Average Tracking Errors (Min. 300 Frames)							
Data Set	TTACNN	ADT	MTUMT	VTACDT	APGCF	AWSODM	Proposed Algorithm
Blur Car	0.46	0.42	0.21	0.17	0.055	0.21	0.041
Running Dog	0.059	0.057	0.48	0.056	0.051	0.061	0.048
Vehicle at Night	0.09	0.088	0.099	0.094	0.071	0.041	0.012
Grayscale vehicle	0.10	0.101	0.118	0.089	0.09	0.088	0.079
Singer	0.14	0.14	0.211	0.1328	0.129	0.144	0.127

The comparison between different state-of-the-art recent algorithms was performed in terms of execution time (in seconds), average tracking errors, precision, mean average precision (MAP) and the precision recall for a minimum of 300 frames of five different image processing datasets. The results depicted in Table 1 clearly show that the proposed algorithm performs better in terms of execution time compared to its counterparts. The gradient descent approach manages the convergence of the particles in a fast time compared to the other tracking algorithms, thus providing a better execution time. In Table 2, the proposed algorithm encompasses much fewer average tracking errors compared to its counterparts. The average tracking error is measured by measuring the average deviation of the bounding box from its mean position. Table 3 shows the comparison of algorithms based on precision. Table 4 shows the comparison of the algorithms based on MAP, while Table 5 shows the comparison of the algorithms based on precision recall. All three metrics are precision-based and are measured using the position of the bounding box over the object of interest. Precision is the ratio of the area of overlap to the area of union, and its normalized average value is measured between 0 and 1. MAP is the mean of the average precisions (APs) over a complete dataset. Recall is the metric that specifies the ability of the object detector to successfully detect the object of interest, i.e., the ratio of true positives to the total number of cases. Tables 3, 4 and 5 are the based-on precision parameters and clearly depict the performance of the proposed algorithm was better than its counterparts in terms of precision, MAP and recall. The algorithms are tested on a Corei7 machine in a MATLAB 2019 environment to maintain uniformity.

Table 3. Performance evaluation based on Precision.

Comparison Based on Precision (Min. 300 Frames)							
Data Set	TTACNN	ADT	MTUMT	VTACDT	APGCF	AWSODM	Proposed Algorithm
Blur Car	0.88	0.88	0.93	0.94	0.94	0.92	0.96
Running Dog	0.90	0.92	0.89	0.93	0.94	0.87	0.94
Vehicle at Night	0.91	0.95	0.88	0.92	0.97	0.86	0.98
Gray scale vehicle	0.96	0.96	0.91	0.97	0.99	0.97	1.00
Singer	0.94	0.92	0.95	0.98	0.97	0.98	1.00

Table 4. Performance evaluation based on MAP.

Comparison Based on MAP (Min. 300 Frames)							
Data Set	TTACNN	ADT	MTUMT	VTACDT	APGCF	AWSODM	Proposed Algorithm
Blur Car	69.6	64.8	69.9	73.9	70.1	73.8	74.6
Running Dog	74.0	66.9	71.0	73.1	71.9	71.7	72.9
Vehicle at Night	77.2	68.2	71.1	76.9	74.6	75.1	77.8
Gray scale vehicle	76.1	69.2	72.5	74.9	77.1	77.9	78.2
Singer	74.9	66.0	72.8	75.5	70.9	72.8	75.6

Table 5. Performance evaluation based on Recall.

Comparison Based on Recall (Min. 300 Frames)							
Data Set	TTACNN	ADT	MTUMT	VTACDT	APGCF	AWSODM	Proposed Algorithm
Blur Car	0.55	0.52	0.59	0.53	0.59	0.55	0.52
Running Dog	0.52	0.54	0.54	0.45	0.54	0.49	0.45
Vehicle at	0.46	0.44	0.39	0.46	0.49	0.44	0.41

Night							
Gray scale vehicle	0.49	0.46	0.46	0.42	0.51	0.44	0.40
Singer	0.41	0.38	0.44	0.44	0.39	0.39	0.35

4. Conclusions

The paper presents a tracking routine that tracks the object of interest (such as an object or a subject) using the gradient descent approach. The algorithm detects an object using the MACH filter, which recognizes the object of interest using ASM and generates a correlation peak depicting the presence of an object. A bounding box is constructed around the object once MACH recognizes it. The coordinates of the bounding box are then fed to the gradient descent-based object tracking routine, which tracks the object of interest in successive frames using a step size and the terminal function. The presence of the terminal function enables a much faster convergence of particles in the gradient-based algorithm compared to the conventional state-of-the-art algorithms. The proposed algorithm has a significant scope to improve in the future, as the gradient descent algorithm can be further improved and its ensemble with particle swarm optimization can yield even better convergence results for object and human recognition [49–51]. Moreover, the deep learning based shall be more useful for the recognition task [52–59].

Author Contributions: Conceptualization, H.M., A.Z. and M.U.A.; methodology, M.A.K., H.M., A.Z. and T.H.; software, H.M., T.H. and M.A.K.; validation, M.U.A., R.D., U.T. and A.Z.; formal analysis, M.A.K. and U.T.; investigation, R.D. and M.A.K.; resources, U.T. and R.D.; data curation, H.M. and A.Z.; writing—original draft preparation, T.H., H.M. and A.Z.; writing—review and editing, M.A.K., R.D. and U.T.; visualization, R.D. and M.A.K.; supervision, M.U.A., T.H. and U.T.; project administration, M.A.K., U.T. and R.D.; funding acquisition, R.D. and M.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Urban Lisa dataset is available from <http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm> (accessed on 11 October 2021). The Visual Tracker Benchmark is available from http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (accessed on 11 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kaushal, M.; Khehra, B.S.; Sharma, A. Soft Computing based object detection and tracking approaches: State-of-the-Art survey. *Appl. Soft Comput.* **2018**, *70*, 423–464, <https://doi.org/10.1016/j.asoc.2018.05.023>.
2. Bai, Z.; Li, Y.; Chen, X.; Yi, T.; Wei, W.; Wozniak, M.; Damasevicius, R. Real-Time Video Stitching for Mine Surveillance Using a Hybrid Image Registration Method. *Electronics* **2020**, *9*, 1336, [doi:10.3390/electronics9091336](https://doi.org/10.3390/electronics9091336).
3. Olszewska, J.I. Active contour based optical character recognition for automated scene understanding. *Neurocomputing* **2015**, *161*, 65–71, <https://doi.org/10.1016/j.neucom.2014.12.089>.
4. Mu, H.; Sun, R.; Yuan, G.; Wang, Y. Abnormal Human Behavior Detection in Videos: A Review. *Inf. Technol. Control* **2021**, *50*, 522–545, <https://doi.org/10.5755/j01.itc.50.3.27864>.
5. Zhou, B.; Duan, X.; Ye, D.; Wei, W.; Woźniak, M.; Połap, D.; Damaševičius, R. Multi-Level Features Extraction for Discontinuous Target Tracking in Remote Sensing Image Monitoring. *Sensors* **2019**, *19*, 4855, <https://doi.org/10.3390/s19224855>.
6. Wolf, C.; Lombardi, E.; Mille, J.; Celiktutan, O.; Jiu, M.; Dogan, E.; Eren, G.; Baccouche, M.; Dellandréa, E.; Bichot, C.-E.; et al. Evaluation of video activity localizations integrating quality and quantity measurements. *Comput. Vis. Image Underst.* **2014**, *127*, 14–30, <https://doi.org/10.1016/j.cviu.2014.06.014>.
7. Ge, H.; Zhu, Z.; Lou, K.; Wei, W.; Liu, R.; Damaševičius, R.; Woźniak, M. Classification of Infrared Objects in Manifold Space Using Kullback-Leibler Divergence of Gaussian Distributions of Image Points. *Symmetry* **2020**, *12*, 434, <https://doi.org/10.3390/sym12030434>.
8. Wang, X. Deep Learning in Object Recognition, Detection, and Segmentation. *Found. Trends® Signal Process.* **2014**, *8*, 217–382, <https://doi.org/10.1561/20000000071>.

9. Meyer, F.; Williams, J. Scalable Detection and Tracking of Geometric Extended Objects. *IEEE Trans. Signal Process.* **2021**, *69*, 6283–6298, <https://doi.org/10.1109/tsp.2021.3121631>.
10. Mondal, A. Occluded object tracking using object-background prototypes and particle filter. *Appl. Intell.* **2021**, 1–21, <https://doi.org/10.1007/s10489-020-02047-x>.
11. Liu, T.; Liu, Y. Moving Camera-Based Object Tracking Using Adaptive Ground Plane Estimation and Constrained Multiple Kernels. *J. Adv. Transp.* **2021**, *2021*, 1–15, <https://doi.org/10.1155/2021/8153474>.
12. Demiroz, B.E.; Ari, I.; Eroglu, O.; Salah, A.A.; Akarun, L. Feature-based tracking on a multi-omnidirectional camera dataset. In Proceedings of the 2012 5th International Symposium on Communications, Control and Signal Processing, Rome, Italy, 2–4 May 2012.
13. Masood, H.; Rehman, S.; Khan, A.; Riaz, F.; Hassan, A.; Abbas, M. Approximate Proximal Gradient-Based Correlation Filter for Target Tracking in Videos: A Unified Approach. *Arab. J. Sci. Eng.* **2019**, *44*, 9363–9380, <https://doi.org/10.1007/s13369-019-03861-3>.
14. Wei, W.; Zhou, B.; Maskeliunas, R.; Damaševičius, R.; Połap, D.; Woźniak, M. Iterative Design and Implementation of Rapid Gradient Descent Method. In *Artificial Intelligence and Soft Computing*; Springer: Cham, Switzerland, 2019; pp. 530–539, https://doi.org/10.1007/978-3-030-20912-4_48.
15. Fan, J.; Shen, X.; Wu, Y. What Are We Tracking: A Unified Approach of Tracking and Recognition. *IEEE Trans. Image Process.* **2012**, *22*, 549–560, <https://doi.org/10.1109/TIP.2012.2218827>.
16. Xia, Y.; Qu, S.; Goudos, S.; Bai, Y.; Wan, S. Multi-object tracking by mutual supervision of CNN and particle filter. *Pers. Ubiquitous Comput.* **2019**, *25*, 979–988, <https://doi.org/10.1007/s00779-019-01278-1>.
17. Huyan, L.; Bai, Y.; Li, Y.; Jiang, D.; Zhang, Y.; Zhou, Q.; Wei, J.; Liu, J.; Zhang, Y.; Cui, T. A Lightweight Object Detection Framework for Remote Sensing Images. *Remote Sens.* **2021**, *13*, 683, <https://doi.org/10.3390/rs13040683>.
18. Cui, N. Applying Gradient Descent in Convolutional Neural Networks. *J. Phys. Conf. Ser.* **2018**, *1004*, 012027, <https://doi.org/10.1088/1742-6596/1004/1/012027>.
19. MacLean, J.; Tsotsos, J. Fast pattern recognition using gradient-descent search in an image pyramid. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR-2000), Barcelona, Spain, 3–7 September 2000; <https://doi.org/10.1109/icpr.2000.906213>.
20. Qiu, J.; Ma, M.; Wang, T.; Gao, H. Gradient Descent-Based Adaptive Learning Control for Autonomous Underwater Vehicles With Unknown Uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5266–5273, <https://doi.org/10.1109/tnnls.2021.3056585>.
21. Iswanto, I.A.; Li, B. Visual Object Tracking Based on Mean-shift and Particle-Kalman Filter. *Procedia Comput. Sci.* **2017**, *116*, 587–595, <https://doi.org/10.1016/j.procs.2017.10.010>.
22. Ge, H.; Zhu, Z.; Lou, K. Tracking Video Target via Particle Filtering on Manifold. *Inf. Technol. Control* **2019**, *48*, 538–544, <https://doi.org/10.5755/j01.itc.48.4.23939>.
23. Bhat, P.G.; Subudhi, B.N.; Veerakumar, T.; Laxmi, V.; Gaur, M.S. Multi-Feature Fusion in Particle Filter Framework for Visual Tracking. *IEEE Sens. J.* **2019**, *20*, 2405–2415, <https://doi.org/10.1109/jsen.2019.2954331>.
24. Li, S.; Zhao, S.; Cheng, B.; Chen, J. Dynamic Particle Filter Framework for Robust Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, <https://doi.org/10.1109/tcsvt.2021.3109981>.
25. Malviya, V.; Kala, R. Trajectory prediction and tracking using a multi-behaviour social particle filter. *Appl. Intell.* **2021**, 1–43, <https://doi.org/10.1007/s10489-021-02286-6>.
26. Zhou, Z.; Zhou, M.; Li, J. Object tracking method based on hybrid particle filter and sparse representation. *Multimed. Tools Appl.* **2016**, *76*, 2979–2993, <https://doi.org/10.1007/s11042-015-3211-3>.
27. Lin, S.D.; Lin, J.; Chuang, C. Particle filter with occlusion handling for visual tracking. *IET Image Process.* **2015**, *9*, 959–968, <https://doi.org/10.1049/iet-ipr.2014.0666>.
28. Choe, G.; Wang, T.; Liu, F.; Choe, C.; Jong, M. An advanced association of particle filtering and kernel based object tracking. *Multimed. Tools Appl.* **2014**, *74*, 7595–7619, <https://doi.org/10.1007/s11042-014-1993-3>.
29. Nsing, R.; Karungaru, S.; Terada, K. A comparative study of batch ensemble for multi-object tracking approximations in embedded vision. *Proc. SPIE* **2021**, *11794*, 257–262, <https://doi.org/10.1117/12.2589037>.
30. Li, G.; Liang, D.; Huang, Q.; Jiang, S.; Gao, W. Object tracking using incremental 2D-LDA learning and Bayes inference. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 1568–1571, <https://doi.org/10.1109/icip.2008.4712068>.
31. Dhassi, Y.; Aarab, A. Visual tracking based on adaptive interacting multiple model particle filter by fusing multiples cues. *Multimed. Tools Appl.* **2018**, *77*, 26259–26292, <https://doi.org/10.1007/s11042-018-5852-5>.
32. Shi, Y.; Zhao, Y.; Deng, N.; Yang, K. The Augmented Lagrange Multiplier for robust visual tracking with sparse representation. *Optik* **2015**, *126*, 937–941, <https://doi.org/10.1016/j.ijleo.2015.02.077>.
33. Kong, J.; Liu, C.; Jiang, M.; Wu, J.; Tian, S.; Lai, H. Generalized ℓ_P -regularized representation for visual tracking. *Neurocomputing* **2016**, *213*, 155–161.
34. França, G.; Robinson, D.P.; Vidal, R. Gradient flows and proximal splitting methods: A unified view on accelerated and stochastic optimization. *arXiv* **2019**, arXiv:1908.00865.
35. Chen, J.-X.; Zhang, Y.-N.; Jiang, D.-M.; Li, F.; Xie, J. Multi-class Object Recognition and Segmentation Based on Multi-feature Fusion Modeling. In Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE

- 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 August 2015; pp. 336–339, <https://doi.org/10.1109/uic-atc-scalcom-cbdcom-iop.2015.72>.
36. Majeed, I.; Arif, O. Non-linear eigenspace visual object tracking. *Eng. Appl. Artif. Intell.* **2016**, *55*, 363–374, <https://doi.org/10.1016/j.engappai.2016.08.005>.
37. Assirati, L.; da Silva, N.; Berton, L.; Lopes, A.A.; Bruno, O. Performing edge detection by Difference of Gaussians using q-Gaussian kernels. *J. Phys. Conf. Ser.* **2014**, *490*, 012020. <https://doi.org/10.1088/1742-6596/490/1/012020>.
38. Dinc, S.; Bal, A. A Statistical Approach for Multiclass Target Detection. *Procedia Comput. Sci.* **2011**, *6*, 225–230, <https://doi.org/10.1016/j.procs.2011.08.042>.
39. Bone, P.; Young, R.; Chatwin, C. Position-, rotation-, scale-, and orientation-invariant multiple object recognition from cluttered scenes. *Opt. Eng.* **2006**, *45*, 077203, <https://doi.org/10.1117/1.2227362>.
40. Birch, P.; Mitra, B.; Bangalore, N.M.; Rehman, S.; Young, R.; Chatwin, C. Approximate bandpass and frequency response models of the difference of Gaussian filter. *Opt. Commun.* **2010**, *283*, 4942–4948, <https://doi.org/10.1016/j.optcom.2010.07.047>.
41. Rehman, S.; Riaz, F.; Hassan, A.; Liaquat, M.; Young, R. Human detection in sensitive security areas through recognition of omega shapes using MACH filters. *Proc. SPIE* **2015**, *9477*, 947708, <https://doi.org/10.1117/12.2176841>.
42. Urban Lisa Dataset. Available online: <http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm> (accessed on 11 October 2021).
43. Visual Tracker Benchmark Dataset Webpage. Available online: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (accessed on 11 October 2021).
44. Zhang, L.J.; Wang, C.; Jin, X. Research and Implementation of Target Tracking Algorithm Based on Convolution Neural Network. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018. <https://doi.org/10.23919/chicc.2018.8483713>.
45. Ming, Y.; Zhang, Y. ADT: Object Tracking Algorithm Based on Adaptive Detection. *IEEE Access* **2020**, *8*, 56666–56679, <https://doi.org/10.1109/access.2020.2981525>.
46. Yang, B.; Tang, M.; Chen, S.; Wang, G.; Tan, Y.; Li, B. A vehicle tracking algorithm combining detector and tracker. *EURASIP J. Image Video Process.* **2020**, *2020*, 17, <https://doi.org/10.1186/s13640-020-00505-7>.
47. Bumanis, N.; Vitols, G.; Arhipova, I.; Solmanis, E. Multi-object Tracking for Urban and Multilane Traffic: Building Blocks for Real-World Application. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS), 26–28 April 2021; pp. 729–736, <https://doi.org/10.5220/0010467807290736>.
48. Tian, X.; Li, H.; Deng, H. An improved object tracking algorithm based on adaptive weighted strategy and occlusion detection mechanism. *J. Algorithms Comput. Technol.* **2021**, *15*, <https://doi.org/10.1177/1748302620973536>.
49. Afza, F.; Sharif, M.; Kadry, S.; Manogaran, G.; Saba, T.; Ashraf, I.; Damaševičius, R. A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. *Image Vis. Comput.* **2021**, *106*, 104090.
50. Zhang, Y.-D.; Khan, S.A.; Attique, M.; Rehman, A.; Seo, S., A resource conscious human action recognition framework using 26-layered deep convolutional neural network. *Multimed. Tools Appl.* **2021**, *80*, 35827–35849.
51. Nisa, M.; Shah, J.H.; Kanwal, S.; Raza, M.; Damaševičius, R.; Blažauskas, T. Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features. *Appl. Sci.* **2020**, *10*, 4966.
52. Nasir, I.M.; Yasmin, M.; Shah, J.H.; Gabryel, M.; Scherer, R.; Damaševičius, R. Pearson correlation-based feature selection for document classification using balanced training. *Sensors* **2020**, *20*, 6793.
53. Kadry, S.; Parwekar, P.; Damaševičius, R.; Mehmood, A.; Khan, J.A.; Naqvi, S.R. Human gait analysis for osteoarthritis prediction: A framework of deep learning and kernel extreme learning machine. *Complex Intell. Syst.* **2021**, *11*, 1–19.
54. Sharif, M.I.; Alqahtani, A.; Nazir, M.; Alsubai, S.; Binbusayis, A.; Damaševičius, R. Deep Learning and Kurtosis-Controlled, Entropy-Based Framework for Human Gait Recognition Using Video Sequences. *Electronics* **2022**, *11*, 334.
55. Jabeen, K.; Alhaisoni, M.; Tariq, U.; Zhang, Y.-D.; Hamza, A.; Mickus, A.; Damaševičius, R. Breast Cancer Classification from Ultrasound Images Using Probability-Based Optimal Deep Learning Feature Fusion. *Sensors* **2022**, *22*, 807.
56. Khan, S.; Alhaisoni, M.; Tariq, U.; Yong, H.-S.; Armghan, A.; Alenezi, F. Human Action Recognition: A Paradigm of Best Deep Learning Features Selection and Serial Based Extended Fusion. *Sensors* **2021**, *21*, 7941.
57. Akram, T.; Zhang, Y.-D.; Sharif, M. Attributes based skin lesion detection and recognition: A mask RCNN and transfer learning-based deep learning framework. *Pattern Recognit. Lett.* **2021**, *143*, 58–66.
58. Rashid, M.; Sharif, M.; Javed, K.; Akram, T. Classification of gastrointestinal diseases of stomach from WCE using improved saliency-based method and discriminant features selection. *Multimed. Tools Appl.* **2019**, *78*, 27743–27770.
59. Akram, T.; Gul, S.; Shahzad, A.; Altaf, M.; Naqvi, S.S.R.; Damaševičius, R.; Maskeliūnas, R. A novel framework for rapid diagnosis of COVID-19 on computed tomography scans. *Pattern Anal. Appl.* **2021**, *5*, 1–14.